An Area-efficient Half-row Pipelined Layered LDPC Decoder Architecture

Sabooh Ajaz, Tram Thi Bao Nguyen, and Hanho Lee

Abstract—This paper presents an area-efficient half-row pipelined layered low-density parity check (LDPC) decoder architecture for IEEE 802.11ad applications. The proposed decoder achieves a good tradeoff between throughput and area because of its ability to overcome the low-throughput bottleneck in conventional half-row decoders and the high-complexity bottleneck in fully parallel decoders. Synthesis results using TSMC 40 nm CMOS technology shows much better throughput at 10.84 Gbps and superior area efficiency, compared to previously reported LDPC decoders.

Index Terms—LDPC code, half-row, layered, pipelined, decoder

I. INTRODUCTION

Low-density parity check (LDPC) codes, first developed by Gallager in 1962 [1], are widely adopted in many error correction systems primarily because of the superior bit error rate (BER) and high-throughput potential. Moreover, LDPC codes were proven to be near the Shannon limit when decoded iteratively using a belief propagation decoding algorithm or a sum-product algorithm (SPA), and are capable of parallel processing for high-throughput implementations. Therefore, LDPC codes are suitable choices in many emerging communications standards, such as IEEE 802.11ad, IEEE 802.15.3c, and the next-generation (5G) telecommunications standards. Specifically, millimeter wave (mmWave) gigabit wireless, described by the IEEE 802.11ad Working Group [2], uses LDPC codes as a preferred selection for forward error correction (FEC).

Recently, several studies have been conducted on simplifying the very large-scale integration (VLSI) implementation of LDPC decoders. The LDPC codes constructed for simplified and efficient decoder design are known as structured LDPC codes, including architecture-aware LDPC codes [3] and block-LDPC codes [4]. Based on design approaches by Mansour and Shanbhag [3] and Kim et al. [4], quasi-cyclic LDPC (QC-LDPC) codes, which are composed of sub-matrices, have attracted considerable interest among coding researchers for their design, construction, and structural properties. This is mainly due to their ease of hardware implementation, simpler $H$-matrix structure, and excellent performance over noisy channels when decoded by a message-passing algorithm. In QC-LDPC codes, consecutive rows in a block row of an $H$-matrix (a layer) can be processed independently, since the column weight of each sub-matrix is only 1, at most. In other words, instead of one row, one block row, which is composed of $z$ rows (i.e., $z = 42$ for IEEE 802.11ad), can be updated simultaneously in one clock cycle. QC-LDPC codes have great advantages in terms of hardware complexity for both encoder and decoder. A QC-LDPC encoder can be implemented by simple shift registers with linear complexity. For decoder implementation, the quasi-cyclic structure simplifies wire connection and allows partially parallel decoding, which offers a trade-off between decoding complexity and decoding rate. Furthermore, QC-LDPC codes can provide high error-correction performance comparable to random LDPC codes.
codes. Hence, QC-LDPC code, which is very flexible and can be designed with various code rates, multiple block lengths, and several sub-matrix sizes [5, 6], is another good candidate for high-throughput wireless applications.

The development of multi-gigabit data transmission techniques for 60GHz-band wireless communications systems has necessitated the implementation of high-throughput, low-power LDPC decoder architectures to address the continuing demands for ever-higher data rates [7-13]. To meet these demands, multi-gigabit data transmission rate standards have been proposed, such as IEEE 802.11ad [2] for gigabit wireless local area networks (WLANs) and IEEE 802.15.3c for wireless personal area networks (WPANs). Fully parallel LDPC decoders, which directly map row and column processors together according the Tanner graph interconnection network of the corresponding parity check matrix, can generally provide very high throughput [13, 14] while operating at low clock rates. However, as the parallelism increases, a large number of processing units and wires inhibits efficient design, and results in larger hardware and interconnect complexity. Hence, a partial-parallel architecture is a great alternative in order to reduce routing congestion. In partially parallel decoders, multiple row and column processors share the same processor and the same memory unit. Block-structured LDPC codes and scheduling algorithms proposed for these architectures provide overlapped and reordered row and column processing stages to reduce the processing time, which results in decoding throughput in the range of tens to hundreds of Mbps. Unfortunately, the trade-offs between hardware complexity and throughput are obstacles to applying these architectures in high-throughput applications.

For LDPC decoders, one of the major design challenges is to achieve a better area and throughput tradeoff. An attractive design approach for area reduction is a half-row, parallel layered decoder [11], compared to a fully parallel layered decoder [13, 14]. In this approach, the parity check matrix is split into two nearly-separate halves in the vertical direction. This method not only reduces the wire interconnect complexity between row and column processors but also increases the parallelism in the row processing stage. However, a half-row decoder suffers from the low-throughput problem. In this paper, an area-efficient half-row layered LDPC decoder architecture is presented. This work achieves higher throughput and better area efficiency, compared to other LDPC decoders described in the literature.

The remainder of this paper is organized as follows. In Section II, brief discussions on the pipelined layered LDPC decoding algorithm are carried out. Section III presents the proposed half-row pipelined layered LDPC decoder architecture, and Section IV includes the results of implementation and a comparison. Finally, conclusions are provided in Section V.

### II. Layered LDPC Decoding Algorithm

A belief propagation (BP) algorithm is typically used for decoding LDPC codes, but it requires very high hardware complexity. The conventional BP algorithm consists of check node units (CNUs) and variable node units (VNUs). An interconnection network is also required between CNUs and VNUs to implement proper connections. This type of BP algorithm is also called a two-phase or “flooding” BP algorithm. Another type of algorithm is also described in the literature, called a “layered” decoding algorithm. Compared to the conventional flooding decoding scheme, layered decoding algorithms proposed by Hocevar [15] and by Wang et al. [16] decode using one layer (a row or a block row) at a time. The basic idea is to allow updated information to be utilized more quickly by passing the latest log-likelihood ratios (LLRs) from one layer to another. Use of the most recent LLRs also results in a positive effect on decoding speed, and fewer iterations are required to achieve the final decoded output. Hence, a

<table>
<thead>
<tr>
<th>Algorithm 1: Conventional layered decoding algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initialization:</strong> ( P_i = LLR; R_i = 0 )</td>
</tr>
<tr>
<td><strong>Decoding:</strong></td>
</tr>
<tr>
<td><strong>Check node processing</strong> ( L'_i[k] = L_i[k] - R_i[k - 1] )</td>
</tr>
<tr>
<td>( R_i[k] = \max \min \left{ L'<em>i[k], \beta, 0 \right} \times \prod</em>{n=1}^{m} \text{sign}\left(L'_i[k]\right) )</td>
</tr>
<tr>
<td><strong>Variable node processing</strong> ( P_i[k] = L'_i[k] + R_i[k] )</td>
</tr>
<tr>
<td><strong>Hard decision according to</strong> ( P_i )</td>
</tr>
</tbody>
</table>
layered decoder has the additional advantage of faster convergence. In fact, the layered decoding algorithms [15] for LDPC codes can decrease the number of iterations by almost 50%, compared to the BP decoding algorithm, and they offer almost 50% faster convergence, compared to a conventional flooding LDPC decoder. Therefore, a layered decoding algorithm can offer twice the throughput without any performance degradation. The layered decoding algorithm can be applied horizontally [15, 16] or vertically [17]. A horizontal layered decoding algorithm is much more popular, compared to the vertical layered decoder, since it provides efficient implementation and a great advantage in terms of memory required. The conventional horizontal layered decoding algorithm [15] with the modified Min-Sum algorithm [16] can be described in Algorithm 1. The Min-Sum algorithm is carried out during $I=1, 2, \ldots, I_{\text{max}}$ iterations. Here, let $I$ be the layer number, and let $k$ be the iteration number. Let $R$ denote the check-to-variable message conveyed from check node $c$ to variable node $v$; and let $L$ denote the variable-to-check message from variable node $v$ to check node $c$. In the $k^{th}$ iteration, the LLR message from the $(l-1)^{th}$ layer to the next layer $l$ for variable node $v$ is represented by $P^{l-1}[k]$. The proposed layered LDPC decoder implements Eqs. (4-6) in sequence. In Eq. (5), $N(c)/v$ denotes a set of variable nodes connected to check node $c$, excluding variable node $v$. The iteration process stops when the maximum number of iterations, $I_{\text{max}}$, is reached, or when all parity checks are satisfied. In an LDPC decoder, the CNU is in charge of iterations, and Eq. (3) is dependent upon (1), and Eq. (3) is dependent upon Eq. (2).

Generally, a pipelining technique can be utilized to reduce the critical path in computing units. However, one drawback to layered LDPC decoding is the data dependency between two consecutive layers. This dependency inhibits the layered decoder from pipelining between two layers, which results in a dramatic decrease in performance. If a pipelining technique is adopted among these three consecutive computations without any effective method, the current layer has to wait for the corresponding posterior message updated in the previous layer. Therefore, in order to enable pipelined decoding, an approximate layered decoding scheme was proposed by Kim et al. [4]. Based on this approximation, a pipelined layered LDPC decoding algorithm [4] with one-stage pipelining is formulated in Algorithm 2. This algorithm will serve as a fundamental framework for implementing the proposed half-row pipelined layered LDPC decoder architecture in Section III. Similarly, in the $k^{th}$ iteration, the LLR message from the $(l-2)^{th}$ layer to the next layer $l$ for variable node $v$ is represented by $P^{l-2}[k]$. The proposed layered LDPC decoder implements Eqs. (4-6) in sequence. In Eq. (5), $N(c)/v$ denotes a set of variable nodes connected to check node $c$, excluding variable node $v$. It is clear from Eq. (4) that $L'[k]$ is generated from $P^{l-2}[k]$, unlike the non-pipelined algorithm, which generates $L'[k]$ from $P^{l-1}[k]$. Hence, the latest $R'[k]$ is calculated from $P^{l-2}[k]$. Similarly, $P^{l+1}[k]$ in Eq. (6) is generated from $R^{l-1}[k]$. This one-layer delay in Eqs. (4, 6) enables the insertion of a one-stage pipeline. Eq. (6) does not require the $R$ values of the current layer $(l)$; rather, it takes $R$ values from the previous layer $(l-1)$. As a result, the VNU operation in Eq. (6) is delayed by one clock cycle, which enables the insertion of a one-stage pipeline. This one-layer delay is further exploited to develop a half-row pipelined layered decoder, which is described in the following section.

### III. PROPOSED HALF-ROW PIPELINED LAYERED LDPC DECODER ARCHITECTURE

As mentioned previously, a layered LDPC decoder is mainly the implementation of Eqs. (1-3) of Algorithm 1. The computations given in Eqs. (1-3) are sequentially executed. Eqs. (1, 2) are part of the check node computation, and Eq. (3) is a variable node computation. The conventional layered decoder is shown in Fig. 1(a).
For a layered decoder, both the check node and the variable node computations are combined into what is called check node–based processing. A switch network (SN) is responsible for carrying out LLR permutations based on a value in each $H$-matrix layer. There is no pipeline in a path, apart from input registers. Fig. 1(b) presents a conventional pipelined layered decoder. It is clear that extra complexity is added accordingly since an extra switch network and a subtraction are added. A pipeline is inserted into this decoder to cut the critical path nearly in half. Therefore, it can increase the clock frequency, while clock cycles remain nearly the same. Consequently, throughput is doubled. The one-layer delay described in Section II is exploited to implement an efficient half-row decoder without pipeline delays. The proposed algorithm splits the $H$-matrix into two nearly separate halves. This approach reduces the routing interconnection complexity in the decoder. For instance, the rate-3/4 generator matrix $H$ shown in Fig. 2 is split into two parts, A and B. Each part in a layer (from $L_1$ to $L_4$) is processed separately. The proposed half-row LDPC decoder is shown in Fig. 1(c). Compared to work by Li et al. [11], the proposed half-row pipelined layered LDPC decoder is more efficient in terms of clock cycles since it can process two halves of the $H$-matrix without causing any extra clock cycle delay. Each layer takes two clock cycles, compared to three required by Li et al. [11]. Moreover, feedback is visible after the CNU pipeline to

---

**Fig. 1.** (a) Conventional layered LDPC decoder, (b) conventional pipelined layered LDPC decoder, (c) the proposed half-row pipelined layered LDPC decoder.

---

**Fig. 2.** IEEE 802.11ad rate-3/4 $H$-matrix split in half.

---

**Fig. 3.** Proposed half-row pipelined layered LDPC decoder architecture.
The emerging section split half splits of the one computation in Eq. (6). Output messages from the barrel implements Eqs. (4, 5), whereas the VNU performs the clock cycle VNUs, because they operate on a different layer in each CNU and VNU processing. It is worth mentioning that provided to the barrel shifters, which rotate the LLRs for the LLRs related to Section A sub in CNUs and VNUs. For the 802.11ad standard, each output pipeline stage, one more pipeline stage is inserted and a SN for routing messages. Apart from the input node computation, CNUs for check nodes computation, the proposed decoder mainly consists of VNUs for variable algorithm with a layered decoding algorithm. The pipelined layered LDPC decoder is illustrated in Fig. 3.

Ensure the correct calculation of $L'[k]$.

The detailed architecture of the proposed half-row pipelined layered LDPC decoder is illustrated in Fig. 3. The overall decoder was designed based on the Min-Sum algorithm with a layered decoding algorithm. The proposed decoder mainly consists of VNUs for variable node computation, CNUs for check nodes computation, and a SN for routing messages. Apart from the input and output pipeline stage, one more pipeline stage is inserted in CNUs and VNUs. For the 802.11ad standard, each sub-matrix size is $z = 42$. Therefore, the proposed LDPC decoder consists of 42 parallel CNUs and VNUs in total.

The LLRs related to Section A of the $H$-matrix are provided to the barrel shifters, which rotate the LLRs for CNU and VNU processing. It is worth mentioning that two separate barrel shifter blocks are used for CNUs and VNUs, because they operate on a different layer in each clock cycle due to pipelining. The CNU in Fig. 3 implements Eqs. (4, 5), whereas the VNU performs the computation in Eq. (6). Output messages from the barrel shifters are distributed to CNUs and VNUs, which are one-stage pipelined to facilitate the processing of two half splits of the $H$-matrix concurrently, as shown in Fig. 4. Pipeline sections $XC$, $YC$, $XV$, and $YV$ in Fig. 3 are depicted with the $H$-matrix splits in Fig. 4. For instance, it can be seen from Fig. 4 that during the fifth clock cycle, split-A of layer 3 is processed by section $XC$, while section $YC$ implements split-B of layer 2 and generates the value of $R^2 [k]$ for section $YV$ (VNU layer 2 split-A). The emerging $R^1 [k]$ message generated by the CNU is simultaneously saved in memory and will be given to the VNU as $R^{1-h} [k]$ in the next clock cycle. Similarly, the value of $P^{h-1} [k]$ from the VNU is given to barrel shifter again as $P^{h-1} [k]$ in the next clock cycle. The flip-flop at the end of the minimum sorter in the CNU stores the intermediate minimum result (of split-A) for one clock cycle. The actual minimum value out of the incoming messages for each check node output is valid in the next clock cycle after completion of the split-B processing stage for one layer.

**IV. ANALYSIS AND COMPARISON RESULTS**

The simulation of the proposed half-row pipelined layered LDPC decoder was carried out using binary phase shift keying (BPSK) modulation in an additive white Gaussian noise (AWGN) channel for the IEEE 802.11ad standard. Thereafter, the BER performance of the proposed LDPC decoder for different code rates and a block length of 672 using five-bit quantization is given in Fig. 5. Also, the BER results of original fully parallel LDPC decoder is presented in Fig. 5 to allow for a better comparison. The following labels are used for the figures: “FR” for original fully parallel LDPC decoder and “HR” for the proposed half-row LDPC decoder. Error performance simulation results indicate that the proposed half-row LDPC decoder architecture scheme incurs almost negligible BER loss. It is clearly noticeable that the proposed decoder achieves an acceptable BER with just five iterations, compared to 10 iterations.
proposed by Park et al. [18]. Thus, the proposed decoder has the required BER performance in far fewer iterations, because a layered decoder generally needs fewer iterations for convergence.

The estimated hardware requirements for the proposed half-row pipelined layered LDPC decoder, along with other published LDPC decoders, are summarized in Table 1. In this table, $d_c$ stands for the check node degree, which is equal to 16, 24, and 32, respectively, for IEEE 802.11ad, 802.11n, and 802.15.3c standards. The submatrix size, $z$, is set to 42 for IEEE 802.11ad, to 81 for IEEE 802.11n, and to 21 for IEEE 802.15.3c applications. First of all, analysis results show that the proposed half-row pipelined layered LDPC decoder reduces the complexity of nearly all components by half, compared to the fully parallel pipelined layer LDPC decoder design of Kumawat et al. [19]. Moreover, the proposed decoder reduces the major area-consuming components, like adders, sign magnitude conversion (SMC), and min comparators, compared to flooding decoders [18, 20]. This will lead to a reduction in hardware complexity for the proposed half-row LDPC decoder as these components are the main contributor of logic resources in the decoder architecture. From the above observation, it is clear that the proposed half-row LDPC decoder is more hardware-efficient, compared to other LDPC decoders with a full-row architecture. Although the conventional half-row layered LDPC decoder of Li et al. [11] showed reduced complexity for barrel shifters and subtractions, it consumes many more clock cycles ($3 \times \text{Layers} \times \text{Iterations}$) compared to the proposed decoder ($2 \times \text{Layers} \times \text{Iterations}$).

The proposed LDPC decoder architecture was modeled in the Verilog hardware description language (HDL) and simulated to verify its functionality using a test pattern generated from a C simulator. After completion of the design functionality verification, it was then synthesized using appropriate time and area constraints. Both simulation and synthesis steps were carried out using Synopsys design tools and TSMC 40 nm CMOS technology. Table 2 shows the implementation and performance comparisons between the proposed half-row pipelined layered LDPC decoder and various multi-gigabit state-of-the-art LDPC decoders. Working at 645 MHz, the equivalent gate count of the proposed LDPC decoder is 316 K. This decoder can support four coding rates, i.e. 1/2, 5/8, 3/4, and 13/16, with a code length of 672. The proposed decoder occupies a core area of 0.21 mm² in CMOS 40 nm technology. Although the clock cycles are increased, compared to full-row decoders, the proposed LDPC

![Fig. 5. BER performance of the proposed LDPC decoder for various code rates and a block length of 672 using five-bit quantization.](image-url)

| Table 1. Analysis results for half-row and full-row LDPC decoders |
|---------------------|---------------------|---------------------|---------------------|---------------------|
| IEEE Standard       | 802.11ad           | 802.11ad           | 802.11ad           | 802.11n            |
| Schedule            | Layered            | Layered            | Flooding           | Layered Full-row   |
| Architecture        | Half-row           | Half-row           | Full-row           | 1944               |
| Code length         | 672                | 672                | 672                | 24                 |
| Check node degree $d_c$ | 16                | 16                | 16                | 81                 |
| Sub-matrix size $z$ | 42                 | 42                 | 42                | 21                 |
| Multi-mode          | yes                | yes                | yes               | yes                |
| Estimated Hardware Complexity |  |  |  |  |
| Barrell Shifiters   | $d_c/2 + d_c/2 = 16$ | $d_c/2 = 8$ | $d_c = 16$ | $d_c + d_c = 48$ |
| Adders              | $d_c/2 \times z = 336$ | $d_c/2 \times z = 336$ | $d_c \times z = 672$ | $d_c \times z = 1944$ |
| SMC                 | $d_c/2 \times z = 336$ | $d_c/2 \times z = 336$ | $d_c \times z = 672$ | $d_c \times z = 1944$ |
| Subtractors         | $(d_c/2 + d_c/2) \times z = 672$ | $(d_c/2 + d_c/2) \times z = 672$ | $(d_c + d_c) \times z = 3888$ | $(d_c + d_c) \times z = 672$ |
| Min. Comparator     | $16 \times z = 672$ | $16 \times z = 672$ | $29 \times z = 1218$ | $11 \times z = 891$ |
| Clock Cycles        | $2 \times \text{Layers} \times \text{Iter.}$ | $3 \times \text{Layers} \times \text{Iter.}$ | $- 3 \times \text{Layers} \times \text{Iter.}$ | $- 3 \times \text{Layers} \times \text{Iter.}$ |

The BER performance of the proposed LDPC decoder is shown in Fig. 5.
decoder architecture still achieves the maximum throughput of 10.84 Gbps at 645 MHz, which is more than enough to meet the specifications of the IEEE 802.11ad standard. The half-row LDPC decoder design presented by Li et al. [11] needs about 1.5 times more clock cycles, compared to our proposed decoder. Hence, it requires a much higher clock frequency in order to meet the high-throughput requirement for wireless communications applications. The LDPC decoder by Li et al. [11] achieves only 5.6 Gbps with a maximum clock frequency of 500 MHz for 40 nm CMOS technology. It is obvious that the proposed LDPC decoder architecture achieves better throughput compared to the conventional half-row decoder reported by Li et al. [11]. However, it is quite hard to make a fair comparison between decoders with different code lengths and performance conditions because of their different applications. To take the code length into account, area efficiency (throughput per coded bits per scaled area [21]) is adopted:

\[
\text{area efficiency} = \frac{\text{throughput}}{\text{code length} \times \text{area}}
\]

The corresponding area efficiency of the proposed work is 76.81 Mbps/mm², whereas the area efficiency obtained by Li et al. [11], Park et al. [18], Kumawat et al. [19], and Lee et al. [20] are 52.08 Mbps/mm², 8.37 Mbps/mm², 0.51 Mbps/mm², and 4.41 Mbps/mm², respectively. To give a fair comparison with the decoders that are implemented in different technologies, performance normalization is applied to 65 nm CMOS technology. For Li et al. [11], the scaling factors can be defined as follows:

\[
\text{Norm. area: area} \times (65 \text{ nm/technology})^2
\]
\[
\text{Norm. throughput: throughput} \times (65 \text{ nm/technology}) \times (1/\text{supply voltage})^2
\]
\[
\text{Norm. power: power} \times (65 \text{ nm/technology}) \times (1/\text{supply voltage})^2
\]
\[
\text{Norm. area efficiency: norm. throughput/ (code length} \times \text{norm. area)}
\]
\[
\text{Norm. energy efficiency: norm. power/ bit/ iteration}
\]

Based on the normalized results summarized in Table 2, it can be concluded that the proposed work is the most area-efficient among the reported LDPC decoders. Although the design from Lee et al. [20] achieves the highest data rate (10.97 Gbps at 10 iterations), it suffers from a larger chip core area, compared to our proposed design. Therefore, this results in a lower normalized area efficiency of 11.74 Mbps/mm². The power and energy efficiency results are also presented in Table 2. For fair comparison, the power is normalized by the technology node and supply voltage. The proposed architecture achieves a low normalized energy efficiency of 7.36 pJ/
bit/iteration using a standard 40 nm CMOS technology. Compared with Li et al. [11] and Park et al. [18], our work shows not only higher area efficiency but also better energy efficiency.

Although using different CMOS technology and supply voltage will impact the final results, the proposed half-row pipelined layered LDPC decoder still achieves better performance in terms of throughput and area efficiency, compared to the other conventional LDPC decoders, when normalizing to the same CMOS technology.

V. CONCLUSIONS

This paper presents an area-efficient half-row pipelined layered LDPC decoder, which supports all modes given in the IEEE 802.11ad standard for wireless communications. For rate-3/4, the proposed work presents much higher throughput and area efficiency, compared to other multi-gigabit state-of-the-art LDPC decoders. Furthermore, it also saves considerable area, compared to fully parallel designs. The proposed half-row pipelined layered LDPC decoder architecture is an attractive candidate for next-generation gigabit wireless communications system.

ACKNOWLEDGMENTS

This work was supported by the Basic Science Research Program through the NRF, funded by the Ministry of Science, ICT and Future Planning under Grant 2016R1A2B4015421.

REFERENCES


Sabooh Ajaz received the Bachelor degree in electronic engineering from NED University of Engineering and Technology Karachi, in 2006 and the Master degree from University of Wollongong, Wollongong, Australia, in 2010. He received the Ph.D. degree in Information & Communication Engineering from Inha University, Incheon, Korea in 2015. His research interest lies in VLSI and SoC architecture design for digital signal processing and communication systems.

Tram Thi Bao Nguyen received the B.S degree in Electronic & Telecommunication Engineering from Danang University of Science and Technology, Vietnam, in 2011 and the M.S degree in Information & Communication Engineering from Inha University, Korea in 2016. She is currently working toward the Ph.D. degree in Information & Communication Engineering from Inha University, Korea. Her research interests include VLSI and SoC architecture design for digital signal processing and communication systems.

Hanho Lee received Ph.D. and M.S. degrees, both in Electrical & Computer Engineering, from the University of Minnesota, Minneapolis, in 2000 and 1996, respectively. In 1999, he was a Member of Technical Staff-I at Lucent Technologies, Bell Labs, Holmdel, New Jersey. From April 2000 to August 2002, he was a Member of Technical Staff at the Lucent Technologies (Bell Labs Innovations), Allentown. From August 2002 to August 2004, he was an Assistant Professor at the Department of Electrical and Computer Engineering, University of Connecticut, USA. Since August 2004, he has been with the Department of Information and Communication Engineering, Inha University, where he is currently Professor. From August 2010 to August 2011, he was a visiting scholar at Bell Labs, Alcatel-Lucent, Murray Hill, New Jersey, USA. His research interest includes VLSI architecture design for forward error correction coding, cryptographic, digital signal processing and communications.